Yin, C. et al. (Eds.) (2025). Proceedings of the 6th International conference on Learning Environment and Educational Technology (LEET)

Development of Simulation-based Learning System for Assisting the Novices Learning Recursive Programming Concepts and investigating its learning effects

Pei-Tong CHEN[a]
[a] *Department of Computer Science, University of Taipei, Taipei*
peggiechen0123@hotmail.com
Ah-Fur LAI[b*], **& Cheng-Ying YANG**[c]
[b,c] *Department of Computer Science, University of Taipei, Taipei*
*laiahfur@gmil.com cyang@utaipei.edu.tw

Abstract: Learning programming skills is complex and difficult, especially recursion is an extremely difficult programming and essential concept. Simulation-based learning is to construct a system model or operating program of a real situation through a computer, represent the situations and phenomena that occur in the real world, and provide learners with control and observation. Many research results have found that in simulation-based learning systems can improve learners' problem-solving ability and strengthen motivation; therefore, the main purpose of this study is to develop an online interactive learning system that assists the concept of recursion learning, and to explore the impact of this system on learners' learning achievements and their learning perception towards this system.

Keywords: recursive concepts, simulation-based learning, programming learning, learning achievement

1. Background

The process of programming mainly involves four important steps, including in-depth understanding of problem requirements, formulating solutions, writing code, and testing and debugging. Therefore, programming is a complex process and an abstract concept (Deek, Kimmel, & McHugh, 1998). In programming, learners often encounter difficult to understand concepts such as loops and variables (Topalli & Cagiltay, 2018). Especially, recursion is an extremely difficult concept (Wu, Dale, & Bethel, 1998).

Recursion is an important skill in programming, especially for students studying information technology, it has a significant impact on the performance of subsequent courses, such as data structures and algorithms. When using recursion techniques for merge sort and depth-first search of binary trees, the codes become very concise, but behind the short program is the internal structure of the computer system (i.e., the system stack) and automatic operation, which are difficult for beginners to truly understand, resulting in the learner's misconceptions and learning frustrations. If the internal operation of the system during executing recursion can be presented to beginners, it will help students grasp the concepts of recursion. This is the goal of this research to develop an interactive assisted recursion learning system.

Simulation-based learning has great teaching potential, especially in learning situations that are difficult to achieve in traditional school education (Alhadlaq, 2023). For example, flight simulators and medical simulation tools provide a highly simulated learning experience, allowing learners to learn in situations where they cannot actually perform operations. In the case of simulation, similar real events, objects or phenomena are presented and the results of actions are observed. This simulation-based learning provides learners with valuable experience (Price, 1991; Sharma, 2017). In the field of engineering education, a number of literature studies have shown that the use of simulation-based learning has

significant effects on improving learning outcomes, including enhancing students' understanding of concepts (Fraser et al., 2007) and promoting their learning ability (Chung, Harmon, & Baker, 2001; Ndahi et al., 2007). There are two kinds of educational simulation including operational simulation and conceptual simulation. The operational simulation is designed to facilitate the construction of practical knowledge, such as medical training, and pilot training (Jimoyiannis, 2009). The conceptual simulation is designed to help the learners to build understanding of domain-specific principles as well as problem-solving skills (Clark, 2005), and it can be used for assisting the learners to construct correct recursive concepts.
The main purpose of this study was to design interactive assisted recursion learning system including simulation-based learning modules and on-line test modules. The simulation-based learning modules are combined with recursive codes to gradually present the system operation and internal structures changes, and dynamic text explanation, simultaneously. In other words, the simulation-based learning modules are used for helping programming beginners to trace and understand the detailed process of recursion execution.

2. Research Purpose

The concept of recursion in programming is more abstract and more difficult to learn than recursion in mathematics. Therefore, the main purpose of this research is to develop an online interactive recursion learning system to assist in learning the concepts of recursion, including simulation-based learning modules and online test modules. Those modules hope to reduce the obstacles that students encounter in learning recursive concepts and enhance their learning outcomes through this learning system. In other words, the main objectives of this study were to (1) develop an online interactive learning system that assists the beginners to learn recursive concepts, (2) explore the impact of this system on learners' learning achievements, and (3) Investigate the learning behavior of learners when using this system for their self-regulated learning, (4) explore learners' learning experience of this system.

Figure 1. System architecture and menu

3. The development of an assisted recursive concepts learning system

This study uses web technology to develop an interactive assisted recursive programming learning system. The learning system mainly includes simulation-based learning (SBL for abbrev.) module and online practice/test module. The former provides a total of 7 recursive sub-learning modules, and the latter is an online self-test module, divided into two categories: filling-in-the-blank based on program tracking results and filling-in-the-blank for partial program codes.

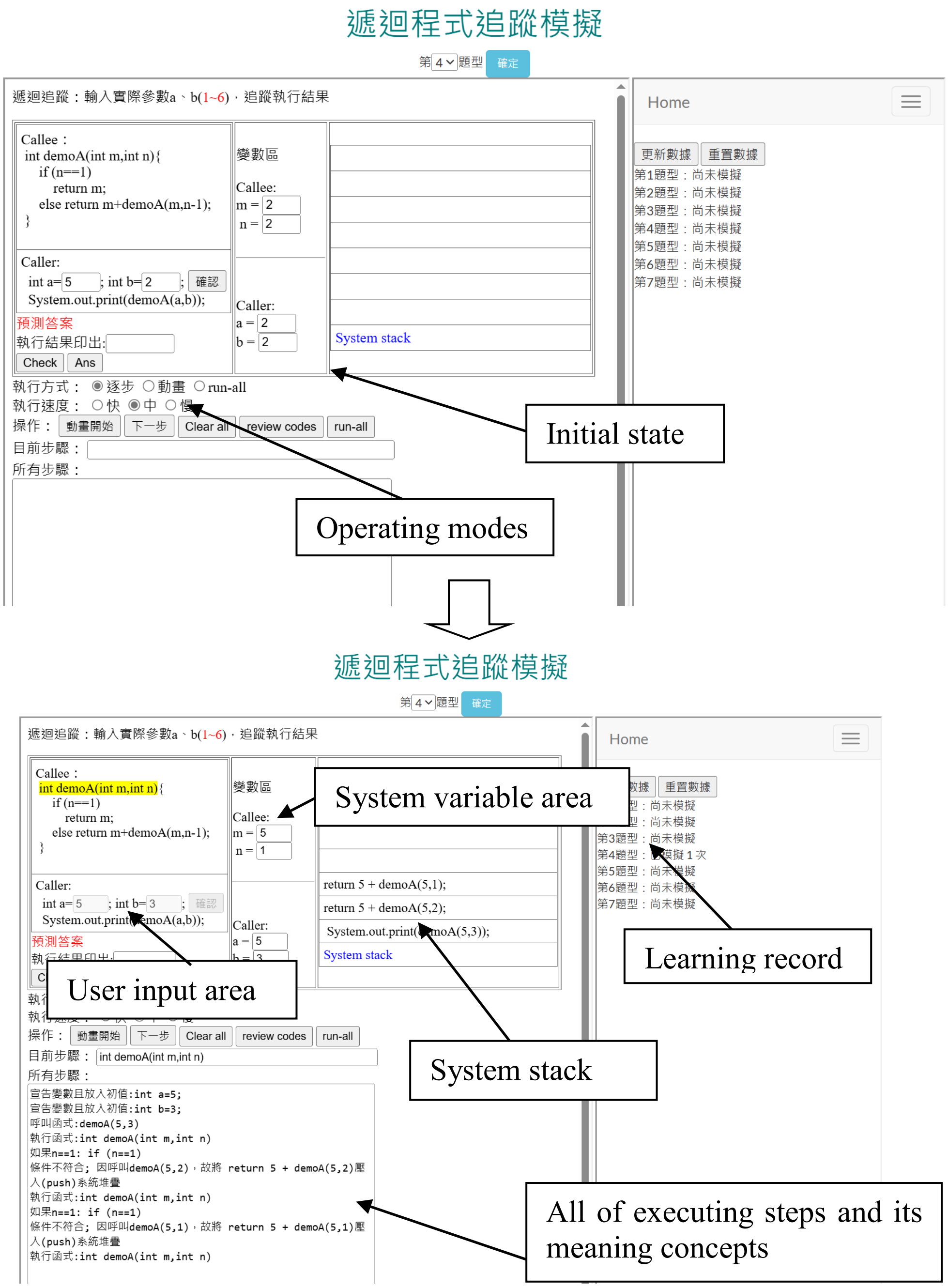

Figure2. Example of simulation-based learning module

The simulation-based learning modules are mainly provided for beginners to track the execution process of recursive codes. It consists of four modes, namely animation mode, step-by-step execution mode, all-at-once presentation mode and prediction result mode. As shown in Figure 2, when beginners select a module, enter the parameter value (such as a, b values) and confirm, then select the execution speed (fast, medium, slow), and then choose the "animation" mode. The system will use animation techniques to gradually execute according to the speed. The code description is added with a background color, explaining the effect of the code description in the "Current Step" area, and also accumulated in the "All Steps" area, while changing the content of the relevant variables of the recursive function (represented by graphics) and the system stack content. In the latter, unfinished statements are pushed onto the stack, and when the statements are completed, the statements on the system stack are popped off and executed. In step-by-step mode, learners are required to press the "Next" button, and the presentation method is the same as the "Animation" mode. The operation is repeated until the recursive function ends in step-by-step mode. In the prediction result mode, learners can input the execution result, and then the interactive system will make a judgment. This mode is similar to the prediction (predict) of the POE strategy (Predict, Observe, Explain). When the answer is judged to be wrong, the learner is motivated to watch the detailed process of simulation-based learning perhaps. This simulation-based learning module provides a visual and process-oriented learning environment to visualize abstract concepts, and help learners build correct and concrete recursive knowledge.

In the online practice module (dill and practice), learners can select the challenge type (recursive program tracking result and partial recursive program code fill-in-the-blank) and test questions by themselves, as shown in Figure 3. The interactive system will judge whether they are correct or not and display the cumulative score.

In terms of self-regulated learning records, learners can review their simulation-based learning times, online test times and scores in both simulation-based learning modules and online practice modules, and these records will be stored in the backend database. This can be used for subsequent analysis (such as Table 3 and Table 4), and students can also fill in the self-study sheet to reflect on their own learning involvement.

Figure 3. Example of online practice/test module

4. Methodology

This study adopted a single-group pre-test and post-test quasi-experimental design for conducting a learning experiment. The subjects are freshmen taking a data structure course. The experiment employed an self-regulated learning strategy, allowing college students to carry out self-regulated learning planning, and learn the recursive concepts in this system. The students can record learning status of the system on their learning sheet, including reflecting on their familiarity with the recursive program, learning confidence, and asking relevant recursive questions. The system will display the number of simulations and tests and their test scores, allowing learners to grasp their own learning status. Before and after participating in the activity, the students took the repetitive learning achievement test before and after the experiment, and then filled out the learning perception toward this activity freely after the experiment.

The recursive learning achievement test items are to mainly examine the ability of tracking recursive program and filling in the blanks of the recursive program. The types of questions in the pre-test and post-test are the same, but the parameter calls are different. The learning achievement test paper has been pre-tested and the analysis results are consistent with the assessment in terms of standard of difficulty index and discrimination index. The learning perception scale was pre-tested, and the collected data were analyzed by factor analysis to obtain three dimensions, including learning benefits of the learning system, self-efficacy, and learning willingness. The overall explanatory power was 75.14%, and the Cronbach Alpha was .932, showing this scale has good construct validity and reliability, so it is suitable for testing.

5. Results

5.1. Results of recursive concepts learning achievement

The mean score of the recursive learning achievement pre-test was 6.30 (SD=3.955), and the mean score of the recursive learning achievement post-test was 8.42 (SD=4.743). The paired sample t test was used, and the result was $t=-4.329$, $p<.001$, reaching a significant level, indicating that the post-test is significantly better than the pre-test, as shown in Table 1; in other words, this interactive assisted learning system can effectively improve students' recursive learning achievements.

In order to explore the progress of learning achievement in different learning achievement groups, paired sample t test was used. The results are shown in Table 2. For the high learning achievement group, $t = -.174$, $p>.05$, which did not reach the significant level, indicating that there was no significant difference between the post-test (M=10.48, SD=4.289) and the pre-test (M=10.32, SD=1.287) in the high learning achievement group.

Table **1.** *The result of pair sample t test of recursive learning achievement*

Test Type	Mean	N	SD	t	Sig.
pretest	6.30	64	3.955	-4.329***	.000
posttest	8.42	64	4.743		

****p<.001*

Table 2. *The result of pair sample t test of recursive learning achievements for different achievement groups*

Group	Test Type	Mean	N	SD	t	Sig.
High	pretest	10.32	22	1.287	-.174	.863
	posttest	10.48	22	4.289		
Middle	pretest	6.91	22	1.109	-7.150***	.000
	posttest	10.07	22	2.532		
Low	pretest	1.20	20	1.473	-3.205**	.005
	posttest	4.35	20	4.657		

***p<.01   ***p<.001*

For the middle learning achievement group, $t=7.150$, $p<.001$, reached a significant level. In the middle learning achievement group, the post-test of the learning achievement (M=10.07, SD=2.532) was significantly better than the pre-test (M=6.91, SD=1.109).
For the low learning achievement group, $t=-3.205$, $p<.01$, reached a significant level. The post-test of the low learning achievement group (M=4.35, SD=4.657) was significantly better than the pre-test (M=1.20, SD=1.473). In other words, this assisted learning system can effectively improve the recursion concepts learning performance of students in the middle and low learning achievement

groups. Above all, the system has the greatest impact on learning achievement of the middle learning achievement group.

5.2. Learning behaviors of the learners in the system

Table 3. *The result of descriptive statistics for different achievement groups' SBL times and self-practice times*

Groups	N	SBL times		Self practice times	
		mean	SD	mean	SD
high	22	10.41	17.773	19.773	35.365
middle	22	13.23	12.486	26.591	45.763
Low	20	12.85	16.420	21.700	32.734
Total	64	12.14	15.498	22.719	38.036

This study defined students' self-regulated learning behavior in this system as the number of times they operate simulation-based learning modules and take online tests. The descriptive statistical results are shown in Table 3. The middle learning achievement group has the highest number of simulation-based learning and tests, indicating that this group was more engaged in self-learning. One-way ANOVA was then used to examine the differences in the number of simulations and tests among the different learning achievement groups. The results are shown in Table 4. The significance levels are all above .05, indicating that there are no significant differences in the number of simulations and tests among these three learning achievement groups.

Table 4. *Results of one-way ANOVA for different achievement groups' SBL times and self-practice times*

behavior	source	SS	df	MS	F	Sig.
SBL times	Between group	102.003	2	51.001	.207	.814
	error	15029.732	61	246.389		
	total	15131.734	63			
Self practice times	Between group	541.556	2	270.778	.182	.834
	error	90601.382	61	1485.269		
	total	91142.938	63			

Table 5. *The result of one-sample t test about learning perception*

Dimensions	N	Mean	SD	$t_{(45)}$
learning benefits of the learning system	46	3.72	.669	7.349***
Self-efficacy	46	3.66	.665	6.762***
Learning willing	46	3.89	.663	9.116***

****p<.001*

5.3. The result of students' learning perception towards this system and activity

The means of the three dimensions of learners' learning perception towards this system and activity were 3.72, 3.66, and 3.89, respectively. The willingness to learn was the highest. A one-sample t test was used, with 3 (representing the average level of agreement) as the test value. The t values are 7.349, 6.762, and 9.116, respectively. The p values are all less than .001, which are all significant. This indicates that the students' feelings about the assisted system and self-regulated learning are significantly higher than the average level, as shown in Table 5. In other words, the students expressed positive learning perception and experience.

6. Conclusions and future study

This study initially analyzed invoices' misconceptions about recursive programming and uses web technology to develop an interactive assisted recursive programming learning system. The learning system includes simulation-based learning modules and online practice modules. The former includes animation mode, step-by-step execution mode, one-time presenting all processes mode, and prediction mode. This system was integrated into university courses for conducting learning experiment by using self-regulated learning strategy. The experimental results show that this learning system and activities can effectively improve the learning achievement of recursive programming concepts, especially for students of the middle learning achievement group. The positive learning effect is the greatest. Students' learning experience of the system and activities was highly positive; in addition, students in different learning achievement groups had similar learning behaviors in the system.
In the future, this study will further analyze the impact of learning systems and activities on the concept of recursive myths, and will add gamification mechanisms to this learning system, such as medals, virtual currency, and system-linked generative AI for recursive code diagnosis, reading, and annotation, based on learning partner requirements.

References

Alhadlaq, A. (2023). Computer-based simulated learning activities exploring Saudi students' attitude and experience of using simulations to facilitate unsupervised learning of science concepts. *Applied Sciences, 13*(7), 4583.

Chung, G. K., Harmon, T. C., & Baker, E. L. (2001). The impact of a simulation-based learning design project on student learning. *IEEE Transactions on Education, 44*(4), 390-398.

Clark, R.C. (2005). Multimedia Learning in e-Courses. In Mayer, R.E., *The Cambridge handbook of multimedia learning* (pp. 589-616). Cambridge: Cambridge University Press.

Deek, F., Kimmel, H., & McHugh, J. A. (1998). Pedagogical changes in the delivery of the first-course in computer science: Problem solving, then programming. *Journal of Engineering Education, 87*(3), 313-320.

Fraser, D. M., Pillay, R., Tjatindi, L., & Case, J. M. (2007). Enhancing the learning of Fluid Mechanics using Computer Simulations. *Journal of Engineering Education, 96*(4), 381-388.

Jimoyiannis, A. (2009). Encyclopedia of Information Communication Technology. DOI: 10.4018/978-1-59904-845-1.ch015

Ndahi, H. B., Charturvedi, S., Akan, A. O., & Pickering, J. W. (2007). Engineering education: Web-based interactive learning resources. *Technology Teacher, 67*(3), 9-14.

Price, R. V. (1991). *Computer-aided instruction: A guide for authors*. Belmont, CA: Wadsworth, Inc.

Sharma, R. (2017). Computer assisted learning-A study. *International Journal of Advanced Research in Education & Technology, 4*(2), 102-105.

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering Education, 120, 64-74. education through problem-based game projects with Scratch. *Computers & Education, 120*, 64-74.

Wu, C.-C., Dale, N. B., & Bethel, L. J. (1998). Conceptual models and cognitive learning styles in teaching recursion. *SIGCSE '98: Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, March 1998, 292-296. https://doi.org/10.1145/273133.274315